

A New Approach to Teaching C++ Programming for Surveyors and Cartographers: Pedagogical Experiment at the Moscow University of Geodesy and Cartography

Vladimir ZABLITSKII, Russia

Key words: teaching C++ programming, training computer programs, direct angular intersection, Gauss formulae

SUMMARY

A new approach to teaching C++ programming for surveyors and cartographers has been developed. Pedagogical experiments were carried out in the period of 2009–2020 in Moscow State University of Geodesy and Cartography. The C++ programming course focuses on the use of cartographic tasks and geodetic exercises to illustrate various programming language constructions. Students have to deal with common geodetic tasks both on lectures and during practical sessions in computer class. This is the main characteristic of the training. The examples of geodetic training programs for first-year students might be the following: determination of the height of a point on the map, calculation of a grid bearing of a line. For the purpose of training students are asked to convert an angle from a radian measure to degrees, minutes, seconds and estimate Gaussian convergence of meridians. The slope of line on the map is to be calculated; the declination of magnetic needle for old topographic map is to be evaluated. During the period of training students also work with programs theodolite and Adrianov's compass and have to program their tasks. Currently, more than fifty training programs are used in the training process. An example the typical training program for students studying the basics of programming in C++ is discussed. The program calculates the flat rectangular coordinates of the desired additional point according to Gauss formulae. The coordinates of the sought point are calculated by the coordinates of two starting points, two angles of the direction of the baselines and two adjacent angles at the base points. The developed program illustrates the calculation of flat rectangular coordinates by means of a straight single angular intersection using procedural programming technology.

A New Approach to Teaching C++ Programming for Surveyors and Cartographers: Pedagogical Experiment at the Moscow University of Geodesy and Cartography (10349)
Vladimir Zablotskii (Russia)

FIG Working Week 2020
Smart surveyors for land and water management
Amsterdam, the Netherlands, 10–14 May 2020

A New Approach to Teaching C++ Programming for Surveyors and Cartographers: Pedagogical Experiment at the Moscow University of Geodesy and Cartography

Vladimir ZABLITSKII, Russia

1. INTRODUCTION

The C++ programming course for undergraduate students has been developed at the Moscow State University of Geodesy and Cartography (MIIGAiK). This course is based on geodetic training tasks and intended to meet the needs of surveying industry where the university plays one of the leading roles. The lectures of this course were illustrated with tasks of general geodesy. Practical lessons at the computer classes and homework are to prepare students to use knowledge of C++ programming methods to solve various engineering and surveying problems. Programming textbooks usually are designed for a wide range of readers so they do not have the programs focused on the training of students specializing in cartography and geodesy. Nevertheless, the list of references contains several generalized textbooks on the C++ language.

Our aim is to develop a set of sample educational geodetic programs and tasks designed to teach C++ programming students making a specialty of geodesy and cartography. Programming of geodetic tasks requires students to have some specific knowledge in geodesy and cartography as well as in programming. Therefore, the programming of geodetic tasks unlike solving typical mathematical problems might train future surveyors and cartographers to solve their specific practical problems. The author argues that effective teaching of programming in the field of engineering should be based on subject-related lectures, textbooks and collections of programming tasks. This task is very urgent at the present time, and it is to be solved promptly with a view to improving the quality of teaching programming in higher education.

Today there are written about seventy training programs on different topics of C++ programming course. These are some examples of geodetic training programs used in course.

Title of the program	What and how the program calculates
Single direct angular intersection on Yung formulas	The coordinates of the unknown point are calculated at the coordinates of the two reference points and the two interior angles of the triangle.
Single direct angular intersection on Gauss formulas	The coordinates of the unknown point are calculated by the coordinates of two starting points, two bearings of the directions of the baselines and two adjacent angles at the base points.
Single angular resection on Delambre formulas	To calculate the coordinates of a point, the coordinates of three reference points and two angles measured at the

A New Approach to Teaching C++ Programming for Surveyors and Cartographers: Pedagogical Experiment at the Moscow University of Geodesy and Cartography (10349)
Vladimir Zablotskii (Russia)

FIG Working Week 2020
Smart surveyors for land and water management
Amsterdam, the Netherlands, 10–14 May 2020

	unknown point are used. Measurements are doing from a defined point.
Single angular resection with an additional checkpoint on Delambre formulas	The coordinates of the four initial points are used to calculate the coordinates of the unknown point, as well as the three angles measured in directions from the starting point to the other three. Measurements are doing from a defined point.
Double direct angular intersection on Yung formulas	The coordinates of the unknown point are calculated at the coordinates of the four reference points and the four interior angles of the two independent triangles.
Double direct angular intersection on Gauss formulas	The coordinates of the unknown point are calculated by the coordinates of four initial points, four grid bearings of the baselines and four adjacent angles of the direction from the base points to unknown point.
Comparison of the effects of curvature of the Earth, the Moon and Mars on measurements of the lengths of horizontal and vertical lines	Calculation of the absolute and relative measurement error of horizontal and vertical line lengths associated with the influence of the curvature of the Earth, the Moon and Mars. Typical dimensions of the surface areas, which could be considered flat having an accuracy of 1:1000000 and 1:200000 for the Moon and Mars, are compared.

2. METHODS OF RESEARCH

The work has been carried out via a personal computer (PC) with Microsoft Windows 8.1 operating system. In this work was used the free open source Code::Blocks environment and GCC C++ compiler. The Code::Blocks/gcc package generates 32-bit programs. The program in this paper runs within a command line prompt and writes out to the command line.

In this paper the curriculum for students specializing in cartography and geodesy and studying the basics of programming in C++ language is considered, demonstrating the solution of direct angular cross-bearing via formulae of tangents of directional angles. We consider the geodetic problem statement. A straight cross-bearing allows defining the coordinates of an additional point using two source points with known coordinates. The solution of straight angular intersection is based on the formulas of tangents of directional angles (Gauss formulas). Consider the coordinates of the initial points $A (X_A, Y_A)$ and $B (X_B, Y_B)$, and there is no visibility between them. It is required to determine the coordinates of point $P (X_P, Y_P)$. Consider two horizontal adjacent angles which have been also measured: angle α between the side of the reference geodetic network and the direction to point P and angle β between another side of the reference geodetic network and the direction to point P (Fig.1).

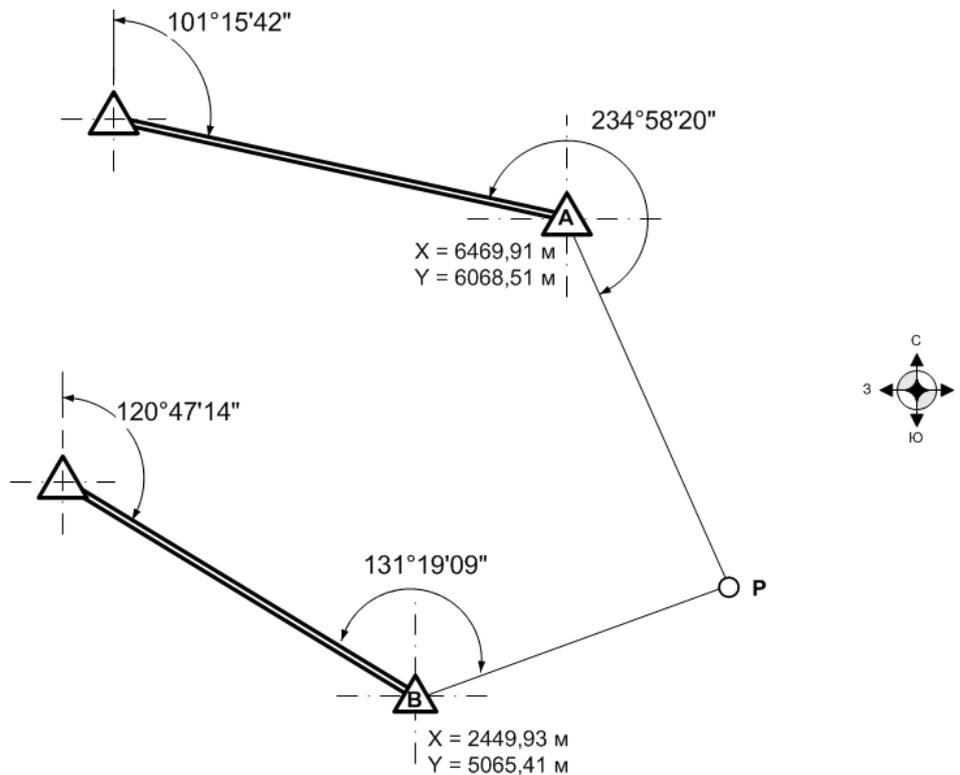


Fig.1. Diagram of straight single angular geodetic intersection via directional angles and left adjacent angles: A, B - initial points, P - sought point

To calculate the x_P and y_P coordinates of a defined point P , it is necessary to use the tangent formulae of the bearing grid angles:

$$x_P = \frac{x_A \cdot \operatorname{tg}\alpha + x_B \cdot \operatorname{tg}\beta - y_A + y_B}{\operatorname{tg}\alpha - \operatorname{tg}\beta}, \quad (1)$$

$$y_P = y_A + (x_P - x_A) \cdot \operatorname{tg}\alpha, \quad (2)$$

where x_A, x_B — the abscissa of points A and B , y_A, y_B — the ordinates of points A and B , $\operatorname{tg}\alpha$ and $\operatorname{tg}\beta$ — the tangents of directional angles from starting points A and B to defining point P .

Bearing grid angles of directions α and β of starting points A and B to defined point P are calculated according to the following formulae:

$$\alpha = \alpha_{AI} + \gamma_{AP}, \quad (3)$$

$$\beta = \beta_{BI} + \gamma_{BP}, \quad (4)$$

where α_{AI} and β_{BI} — the reference angles of the first and second link of the geodetic reference network containing points A and B , γ_{AP} and γ_{BP} — the adjacent angles between the reference

sides of the reference network and the direction to the defined point P . When α and β are being calculated the rule that the left side angles are taken with the plus sign is used, and in case of the right angles they are taken with the minus sign.

3. AN EXAMPLE OF TRAINING PROGRAM

The program developed solves the task of determining unknown coordinates using standard mathematical functions of tangent. Find the listing of the program below:

```

01: #include <iostream>
02: #include <cmath>
03: using namespace std;
04:
05: int main(void)
06: {
07:     double xA, yA, xB, yB, xP, yP;
08:     double degrees, minutes, seconds, angle, azimuth_AP, azimuth_BP;
09:
10:     cout<<"Enter the X, Y-coordinates of the point A: ";
11:     cin >> xA >> yA;
12:
13:     cout <<"Enter the grid azimuth of the line (A-1),
13:             degrees, _, minutes, _, seconds: ";
14:     cin >> degrees >> minutes >> seconds;
15:     angle = degrees + minutes/60 + seconds/3600;
16:     double gridAzimuthOfLineA1 = angle * M_PI/180;
17:
18:     cout <<"Enter the adjoined angle into point A,
18:             degrees, _, minutes, _, seconds: ";
19:     cin >> degrees >> minutes >> seconds;
20:     angle = degrees + minutes/60 + seconds/3600;
21:     double abuttingAngleA = angle * M_PI/180;
22:
23:     azimuth_AP = gridAzimuthOfLineA1 + abuttingAngleA;
24:
25:     cout<<" Enter the X, Y-coordinates of the point B: ";
26:     cin >> xB >> yB;
27:     cout <<" Enter the grid azimuth of the line (B-2),
27:             degrees, _, minutes, _, seconds: ";
28:     cin >> degrees >> minutes >> seconds;
29:     angle = degrees + minutes/60 + seconds/3600;
30:     double gridAzimuthOfLineB1 = angle * M_PI/180;
31:
30:     cout <<"Enter the adjoined angle into point B,
30:             degrees, _, minutes, _, seconds: ";
31:     cin >> degrees >> minutes >> seconds;
32:     angle = degrees + minutes/60 + seconds/3600;
33:     double abuttingAngleB = angle * M_PI/180;
34:
35:     azimuth_BP = gridAzimuthOfLineB1 + abuttingAngleB;
36:
37:     xP = (xA * tan(azimuth_AP) - xB * tan(azimuth_BP) + yB -

```

A New Approach to Teaching C++ Programming for Surveyors and Cartographers: Pedagogical Experiment at the Moscow University of Geodesy and Cartography (10349)
Vladimir Zablotskii (Russia)

FIG Working Week 2020
Smart surveyors for land and water management
Amsterdam, the Netherlands, 10–14 May 2020

```

37:                                     yA) / (tan(azimuth_AP) - tan(azimuth_BP));
38:   yP = yA + (xP - xA) * tan(azimuth_AP) ;
39:
40:   cout <<"X-coordinate of the point P is: " << xP <<" m"<< endl;
41:   cout <<"Y-coordinate of the point P is: " << yP <<" m"<< endl;
42:
43:   return 0;
44:}

```

4. DISCUSSIONS

The training process usually involves the launch of the compiled program on the PC and the subsequent analysis of the program instructions. An interactive electronic board is used. Code analysis with visualization of control flow when calling functions and returning values is usually performed by the student. As programming skills develop, students have the task to analyze the program code orally. Naturally, the initial programs contain only a few lines. Then the number of lines in a code curriculum increases. Oral code analysis involves explaining each line of code and answering the questions what was its purpose and what the given instruction line does.

Then the explanation of what the program does and how it does it is given. This is an example of a guide for the program. The text is put into a methodological guide for students learning the C++ programming language.

Let's consider this code. The main function *main* appears in lines 05-44. The main function defines the variables x_A , y_A , x_B , y_B for storing the planned coordinates of the initial points, and the variables x_P , y_P are for the coordinates of the defined point calculated using the Gauss formulae. The *degrees*, *minutes*, *seconds* variables of the *double* type are also used to enter horizontal angles separately in degrees, minutes, and seconds. The *azimuth_AP* variable is intended for the values of the bearing grid of the direction *AP*, a straight line connecting point *A* and point *P*. The *azimuth_BP* variable is intended for the values of the bearing grid of the direction *BP*, a straight line connecting point *B* and point *P*.

The *main* function contains all the necessary data for calculation. In lines 15-16, the bearing grid angle value (the link of the reference geodetic network where point *A* is located) is converted to a degree measure, in the form of integer and fractional parts of a degree, and then in the angle value appears in radians. The resulting value is stored in the *gridAzimuthOfLineA1* variable. Then, in lines 18-19, the *abuttingAngleA* value of point *A* is to be entered, which is also converted to radians (line 21). And here it is necessary to use the graphic scheme of direct geodetic intersection and follow the rule that for the left along the course of the adjacent corners is taken the sign "plus", and for the right the sign "minus" is taken.

As it can be seen from the diagram in our example, we use left corner angles, that is, we set them with the sign "+", which, however, is omitted when entering angle values separately in degrees minutes and seconds. However, if the right angle was "right", we would enter the angle values in the format $-234^{\circ} -58' -20''$. It is emphasized that each individual value of degrees, minutes and seconds would be entered with the sign "-". The bearing grid of the *AP* direction is calculated in line 23.

The same has to be performed for point *B* in lines 25-35: the value of the bearing grid of *gridAzimuthOfLineB1* of the link of the reference geodetic network in which point *B* is located are to be input from the keyboard, so is the adjacent angle *abuttingAngleB*, and the bearing grid of the *BP* direction is calculated. Calculation of Gauss formulae is carried out in lines 37-38. The results of the determination of the X, Y-coordinates of an additional point *P* are displayed in lines 40-41.

It has to be noted that in the calculation scheme and therefore in the program only formulae of tangents of bearing grid were used, but not formulae of cotangents. This is due to the fact that by means of the computer experiment it has been stated that the values obtained by the formulae of cotangents coincide with the values obtained by the formula of tangents. Even in cases where one of the bearing grid angles of directions to the additional point was of 89°59'59", the calculated values obtained according to the formulae of tangents and cotangents were the same.

Consider the data presented in the following table (taken from Poclad, Gridnev, 2007).

Table 1. Test data for calculating the coordinates of the additional desired point

Point	Bearing grid angle of the geodetic network line	Adjacent angle	Abscissa	Ordinate
A	101°15'42"	234°58'20"	6469,91	6068,51
B	120°47'14"	131°19'09"	2449,93	5065,41

As the result the following will be displayed:

X-coordinate of the point P is: 3233.88 m
Y-coordinate of the point P is: 7493.49 m

Running a program consisted of control data is an important condition when it comes to evaluation of student performance. Both skills of the C ++ programming developed at the course and knowledge of general geodesy and cartography, are to be evaluated.

5. CONCLUSIONS

The experience of teaching C++ programming language for students of cartographers and surveyors at the Moscow University of Geodesy and Cartography since 2009 to 2020 has been generalized. A new course of C++ programming for surveyors and cartographers has been developed. A wide range of training programs with the geodesy and cartography contents has been developed. As example the training program for students studying the basics of C++ programming is discussed. The program demonstrates solution of the straight angular intersection. The developed program calculates the flat rectangular coordinates of the desired additional point according to Gaussian formulae. The coordinates of the sought point are calculated by the coordinates of two starting points, two angles of the direction of the baselines and two adjacent angles at the base points. The program code consists of less than 50 lines. The program demonstrates application of the mathematical tangent function, the

transformation of the angle expressed separately in degrees, minutes and seconds into an angle in degrees with fractional part and further into radians. As a result of the calculation, the program displays the X, Y-coordinates of the desired point — abscissa and ordinate. The developed program illustrates the solution of the problem of calculation of flat rectangular coordinates of the desired point by means of a straight single angular intersection according to Gauss formulae using procedural programming technology.

REFERENCES

Liberty J, Cashman M (2002) SAMS Teach Yourself C++ in 10 Minutes, 2nd Edition, Published by SAMS, 800 East 96th Street, Indianapolis, Indiana, USA.

Prata S (2011) C++ Primer Plus, 6th Edition, Published by Addison–Wesley, Reading, MA, USA.

Poclad G. G., Gridnev S.P. (2007) Geodesy: a textbook for universities. Published by Academic Project, Moscow, Russia.

Stroustrup B. (1997). The C++ Programming Language. Special Edition, Published by Addison-Wesley, Reading, MA, USA.

BIOGRAPHICAL NOTES

Vladimir R. Zablotskii (b. 1953) graduated from the Lomonosov Moscow State University (soil–science faculty in 1976 and physics faculty in 1985). He has the degree of Candidate Science (Biology). He is now the Associate Professor on department of applied cosmonautics and photogrammetry at Moscow University of Geodesy and Cartography and Associate Professor on department of the Physics at Bauman Moscow State Technical University. He is author of more than 50 publications in the field of physics of soils, remote sensing and GIS, application of computer technologies in education and C++ programming for cartographers and geodesist.

CONTACTS

Vladimir Zablotskii
Moscow State University of Geodesy and Cartography,
Gorokhovskiy pereulok 4,
Moscow, 105064
RUSSIA
Tel. +7 9163801461
Email: zablotskii@miiigaik.ru

A New Approach to Teaching C++ Programming for Surveyors and Cartographers: Pedagogical Experiment at the Moscow University of Geodesy and Cartography (10349)
Vladimir Zablotskii (Russia)

FIG Working Week 2020
Smart surveyors for land and water management
Amsterdam, the Netherlands, 10–14 May 2020